



UNITED STATES PATENT AND TRADEMARK OFFICE

HN
UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/731,515	12/09/2003	Marcin Sawicki	60001.295US01	3732
27488	7590	12/29/2005	EXAMINER	
MERCHANT & GOULD (MICROSOFT) P.O. BOX 2903 MINNEAPOLIS, MN 55402-0903			BOTTs, MICHAEL K	
			ART UNIT	PAPER NUMBER
			2176	

DATE MAILED: 12/29/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/731,515	SAWICKI ET AL.	
	Examiner	Art Unit	
	Michael K. Botts	2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 12/9/03; 9/7/04; 11/8/04.
 2a) This action is **FINAL**. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-23 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-23 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on 09 December 2005 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
 3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
 Paper No(s)/Mail Date 9/7/04
- 4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date. _____.
 5) Notice of Informal Patent Application (PTO-152)
 6) Other: _____

DETAILED ACTION

1. This document is the first Office Action on the merits. This action is responsive to the following communications: The Non-Provisional Application, which was filed on December 9, 2003 as a continuation in part of Non-Provisional Application 10/187,060, which was filed on June 28, 2002, and an Information Disclosure Statement (IDS), which was filed on September 7, 2004, and a Preliminary Amendment, which was filed on November 8, 2004.
2. Claims 1-23 have been examined, with claims 1, 11, and 19 being the independent claims.
3. The Drawings are objected to.
4. The Abstract is objected to.
5. The Specification is objected to.
6. Claims 2, 3, 4, 14, 15, 21, and 22 are objected to.
7. Claims 1-23 are rejected.

Information Disclosure Statement

8. An initialed and dated copy of applicant's IDS form 1449, which was filed on September 7, 2004, is attached to this Office Action.

Preliminary Amendment

9. The Preliminary Amendment, which was filed on November 8, 2003, is accepted.

Drawings

10. The drawings are objected to because it is unclear which drawing elements are being identified by the reference characters and lead lines from the following reference characters: 310, 320, 330, 410, 420, 430, 440, and 450.

11. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because reference character "440" has been used to designate two different drawing elements.

12. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because reference character "410" has been used to designate both "instruction 410" and fldChar 410. See, disclosure, page 9, lines 17, and 19.

13. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description: 300, 400, and 500.

14. In response to the above identified objections, corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application. Any

amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Abstract of the Disclosure

The abstract of the disclosure is objected to because of the use it does not accurately reflect the invention claimed. The statement that the invention "may be manipulated on a server or anywhere even when the application creating the ML document is not present" is not claimed, and is essentially inherent in the markup language itself. In addition, the statement that the invention fields "may be manipulated when the ML document is parsed by other applications," similarly identifies a property of a markup language, rather than that of the invention itself. Finally, the statement that "field definition information (i.e. properties) are save in a markup language (ML) document without data loss, while allowing the filed structures to be parsed by ML-aware applications and to be read by ML programmers" also merely states inherent properties of the markup language, rather than stating a concise description of the invention. Correction is required. See MPEP § 608.01(b).

15. Applicant is reminded of the proper content of an abstract of the disclosure.

A patent abstract is a concise statement of the technical disclosure of the patent and should include that which is new in the art to which the invention pertains. If the patent is of a basic nature, the entire technical disclosure may be new in the art, and the abstract should be directed to the entire disclosure. If the patent is in the nature of an improvement in an old apparatus, process, product, or composition, the abstract should include the technical disclosure of the improvement. In certain patents, particularly those for compounds and compositions, wherein the process for making and/or the use thereof are not obvious, the abstract should set forth a process for making and/or use thereof. If the new technical disclosure involves modifications or alternatives, the abstract should mention by way of example the preferred modification or alternative.

The abstract should not refer to purported merits or speculative applications of the invention and should not compare the invention with the prior art.

Where applicable, the abstract should include the following:

- (1) if a machine or apparatus, its organization and operation;
- (2) if an article, its method of making;
- (3) if a chemical compound, its identity and use;
- (4) if a mixture, its ingredients;
- (5) if a process, the steps.

Extensive mechanical and design details of apparatus should not be given.

16. Applicant is reminded of the proper language and format for an abstract of the disclosure.

The abstract should be in narrative form and generally limited to a single paragraph on a separate sheet within the range of 50 to 150 words. It is important that the abstract not exceed 150 words in length since the space provided for the abstract on the computer tape used by the printer is limited. The form and legal phraseology often used in patent claims, such as "means" and "said," should be avoided. The abstract should describe the disclosure sufficiently to assist readers in deciding whether there is a need for consulting the full patent text for details.

The language should be clear and concise and should not repeat information given in the title. It should avoid using phrases which can be implied, such as, "The disclosure concerns," "The disclosure defined by this invention," "The disclosure describes," etc.

The Specification

17. Applicant is required to update the status (pending, allowed, etc.) of all parent priority applications in the first line of the specification. The status of all citations of U.S. filed applications in the specification should also be updated where appropriate.

18. This application contains a computer program listing of more than three hundred (300) lines. In accordance with 37 CFR 1.96(c), a computer program listing contained on more than three hundred (300) lines, must be submitted as a computer program listing appendix on compact disc conforming to the standards set forth in 37 CFR 1.96(c)(2) and must be appropriately referenced in the specification (see 37 CFR 1.77(b)(5)). Accordingly, applicant is required to cancel the current computer program listing, file a computer program listing appendix on compact disc in compliance with 37 CFR 1.96(c), and insert an appropriate reference to the newly added computer program listing appendix on compact disc at the beginning of the specification.

19. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Claims Objections

20. Claim 2 is objected to because of the following informalities: Because the claim reads on a decision of determining a field type from between two mutually exclusive choices, it is believed that the applicants intended the word “and” in line 2 of the claim to be the word “or,” and the claim will be read with the word “or” for the remainder of this Office Action. Appropriate correction is required.

21. Claims 3, 4, 14, 15, 21, and 22 are objected to because of the following informalities: The term “richly formatted text” is not expressly defined in the application. From the context of the use of the term in disclosure and in the claims, it is believed that applicants intended to refer to Microsoft Corporation’s Rich Text Format (RTF) standard, and the claims will be read that “richly formatted text” is the equivalent of Rich Text Format (RTF) for the remainder of this Office Action. Appropriate correction is required.

Claims Rejections – 35 U.S.C. 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

22. Claims 1, 7, 9, and 10 are rejected under 35 U.S.C. 102(b) as being clearly anticipated by Ayers, I., "AbiWord's Potential," Linux Gazette, Issue 43, July 1999, last downloaded by the Examiner on December 20, 2005, from:
www.linuxgazette.com/issue43/ayers.html, downloaded pages 1-4, [hereinafter "Ayers"].

Regarding **independent claim 1**, Ayers teaches:

A method for representing field structures in a markup language document, comprising:
determining properties corresponding to a field that relates to at least one section of an application document;

(It is noted that the term "properties" is not specifically defined in the specification. The disclosure associates the term in a general sense with simple and complex fields, as in the following: "the properties of the complex field (when the field is a complex field) are mapped into elements, attributes, and values of the ML file." Disclosure, page 18, lines 9-10. Further evidence that the term refers to the general appearance of the document is taken from the disclosure, stating: "the fields and the properties associated with the fields may change from page to page, section to section, chapter to chapter and the like." Disclosure, page 18, lines 22-23. Based on the above analysis, it is believed that the applicants intended the term "properties" to be used in the general sense of the appearance of the text, irrespective of the font, and such definition will be used for the remainder of this Office Action. In support of the above definition of "properties" as

known to one of ordinary skill in the art at the time of the invention, see, Harold, Elliotte Rusty, "XML Bible," IDG Books Worldwide, 1999, pages 369-388.)

mapping the properties of the field into at least one of a markup language element, an attribute, and a value; and

(See, Ayers, page 3, last two full paragraphs and citing code example from the screenshot and showing font and line properties and values. Note that Ayers teaches that the code is the markup language mapped from the properties of the document field displayed in the screenshot.)

storing the properties of the field in the markup language document.

(See, Ayers, page 3, last two full paragraphs and citing code example from the screenshot. Note that Ayers teaches saving the document shown in the screenshot, which creates the saved code that is saved in memory.)

Regarding **dependent claim 7**, Ayers teaches:

*The method of Claim 1, further comprising:
determining properties corresponding to an additional field that relates to at least one section of the application document;
mapping the properties of the additional field into at least one of a markup language element, an attribute, and a value; and
storing the properties of the additional field in the markup language document.*

(It is noted that this claim is read as the method including to edit and add or insert text to a document and then to convert to XML and save the resulting combined document.

See, Ayers, page 3, last two lines, teaching the editing, or modification, of an XML document in AbiWord.)

Regarding **dependent claim 9**, Ayers teaches:

The method of Claim 1, wherein the properties of the fields stored in the markup language document are understood by an application that understands the markup language when the field is not native to the application.

(See, Ayers, page 2, third full paragraph, teaching that an AbiWord document is saved in an *.abw file written in XML and that the files can be read by any text editor.)

Regarding **dependent claim 10**, Ayers teaches:

The method of Claim 1, wherein the markup language document is manipulated on a server to substantially reproduce the field of the application document notwithstanding the presence of an application that generated the markup language document.

(See, Ayers, page 2, third full paragraph, teaching that an AbiWord document is saved in an *.abw file written in XML and that the files can be read by any text editor. See also, Ayers, page 3, bottom two lines, teaching that the document save in AbiWord may be modified by hand in an editor rather than a word processor. It is inherent from the

ability to be read by any text editor and be modified by hand that the document may be manipulated on any suitable computing device, including a server.)

23. It is noted that any citations to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art.

See, MPEP 2123.

Claims Rejection – 35 U.S.C. 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

24. Claims 2-6, 8, and 11-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ayers, I., "AbiWord's Potential," *Linux Gazette*, Issue 43, July 1999, last downloaded by the Examiner on December 20, 2005, from: www.linuxgazette.com/issue43/ayers.html, [hereinafter "Ayers"], in view of W3C, "XML Schema Part 0: Primer, W3C Recommendation, 2 May 2001," last downloaded by the Examiner on December 19, 2005, from: www.w3.org/TR/201/REC-xmlschema-0-20010502, downloaded pages 1-67, [hereinafter "XML Schema"], and further in view of

W3C, "XML Schema Requirements, W3C Note 15 February 1999," last downloaded by the Examiner on December 19, 2005, from: www.w3.org/TR/NOTE-xml-schema-req, downloaded pages 1-5, [hereinafter "XML Requirements"].

Regarding **dependent claim 2**, Ayers in view of XML Schema and further in view of XML Requirements teaches:

The method of Claim 1, further comprising determining whether the field is one of a complex field and a simple field.

(See, Ayers, page 3, screenshot and the last two full paragraphs, wherein the screenshot teaches both simple and complex fields, with simple and complex elements, and the code example teaches that the simple field is mapped and stored. Ayers teaches the screenshot with the text to be saved, and Ayers teaches code generated in XML from saving the text in AbiWord. Ayers does not expressly teach to determine whether the field is one of a complex field and a simple field.

XML Schema teaches that simple and complex data elements are defined as part of the XML language. See, XML Schema, downloaded page 7.

Ayers and SML Schema are analogous art because they are from the same field of endeavor of creating and manipulating electronic documents. At the time the invention was made, it would have been obvious to one of ordinary skill in the art to differentiate between a simple element and a complex element, and to use such differentiation to determine how to render a word document. It is at least obvious, and

possibly inherent, from the screenshot and the code example shown in Ayers that AbiWord determines whether the data is complex or simple.

The suggestion or motivation for combining the teachings of Ayers, that a word processor program could convert and store documents in XML, with the teaching of XML Schema, including that document types are distinguished as simple or complex, is expressly stated in XML Requirements, stating: "The following usage scenarios describe XML applications that should benefit from XML schemas. * * * 4. Traditional document authoring/editing governed by schema constraints." See, XML Requirements, page 3.

The combination of the teachings of Ayers to convert and store word processing documents in XML combined with the teachings of SML Schema that data may be one of two types, simple or complex, would be an invention whereby a document mapped to XML would also distinguish and map simple and complex elements, and that such could be stored in memory.)

Regarding **dependent claim 3**, Ayers in view of XML Schema and further in view of XML Requirements teaches:

The method of Claim 2, wherein an instruction portion of the field comprises at least one of richly formatted text and embedded additional fields when the field is a complex field.

(See, Ayers, page 2, third full paragraph, teaching that AbiWord can be saved in RTF format.)

Regarding **dependent claim 4**, Ayers in view of XML Schema and further in view of XML Requirements teaches:

The method of Claim 2, wherein an instruction portion of the field excludes richly formatted text and embedded additional fields when the field is a simple field.

(It is noted that the term “richly formatted text” is not defined in the application. It is believed that applicants intended to refer to Microsoft Corporation’s Rich Text Format (RTF) standard, and this definition will be used for the remainder of this Office Action. It is inherent in the XML schema that simple types cannot have element content and cannot carry attributes, and it is therefore inherent in AbiWord, as taught by Ayers to save documents in XML. See in support of this inherent property, XML Schema, downloaded page 7.)

Regarding **dependent claim 5**, Ayers in view of XML Schema and further in view of XML Requirements teaches:

The method of Claim 2, further comprising representing the field with a fldSimple element when the field is determined to be a simple field.

(It is noted that the term “fldSimple” is not expressly defined in the application. It is believed that the applicants created the term “fldSimple” as an element type. It is further noted that the ability to create an element type is inherent in the XML language and is therefore inherent in AbiWord, which is taught by Ayers to save documents in XML.)

Regarding **dependent claim 6**, Ayers in view of XML Schema and further in view of XML Requirements teaches:

The method of Claim 2, further comprising representing the field with at least one of a fldChar element and instrText element when the field is determined to be a complex field.

(It is noted that the ability to create an element type is inherent in the XML language and is therefore inherent in AbiWord, which is taught by Ayers to save documents in XML.)

Regarding **dependent claim 8**, claim 8 incorporates substantially similar subject matter as claimed in claim 1, and in further view of the following, is rejected along the same rationale.

It is noted that this claim reads on editing a document and then to converting it to XML and saving the resulting combined document, with the addition of converting any newly added text to XML.

See, Ayers, page 3, last two lines, teaching the editing, or modification, of an XML document in AbiWord.

Ayers does not expressly teach determining whether properties associated with all fields of the application document have been stored in the markup language document and further processing any unprocessed fields.

The Examiner takes official notice of the fact that word processor programs are well known to include the function of saving a document periodically and in final form.

For example, Microsoft Word default periodic auto-archive or save file functions. It would have been obvious to one of ordinary skill in the art at the time of the invention to save any edited versions of the document for versioning, archiving, security against loss from sudden computer failure, completion of the document, etc.

Further, it is inherent that a word processing program that saves documents in XML will also have the function to save any additional data added to that document in XML.

Regarding **independent claim 11**, Ayers in view of XML Schema and further in view of XML Requirements teaches:

A computer-readable medium for representing fields in a markup language document, comprising:

determining properties relating to one or more fields used within a word-processing document;

(It is noted that the term “properties” is not specifically defined in the specification. The disclosure associates the term in a general sense with simple and complex fields, as in the following: “the properties of the complex field (when the field is a complex field) are mapped into elements, attributes, and values of the ML file.” Disclosure, page 18, lines 9-10. Further evidence that the term refers to the general appearance of the document is taken from the disclosure, stating: “the fields and the properties associated with the fields may change from page to page, section to section, chapter to chapter and the like.” Disclosure, page 18, lines 22-23. Based on the above analysis, it is believed that

the applicants intended the term “properties” to be used in the general sense of the appearance of the text, irrespective of the font, and such definition will be used for the remainder of this Office Action. In support of the above definition of “properties” as known to one of ordinary skill in the art at the time of the invention, see, Harold, Elliotte Rusty, “XML Bible,” IDG Books Worldwide, 1999, pages 369-388.)

determining whether the field is one of a complex field and a simple field;
(See, Ayers, page 3, screenshot and the last two full paragraphs, wherein the screenshot teaches both simple and complex fields, with simple and complex elements, and the code example teaches that the simple field is mapped and stored. Ayers teaches the screenshot with the text to be saved, and Ayers teaches code generated in XML from saving the text in AbiWord. Ayers does not expressly teach to determine whether the field is one of a complex field and a simple field.

XML Schema teaches that simple and complex data elements are defined as part of the XML language. See, XML Schema, downloaded page 7.

Ayers and SML Schema are analogous art because they are from the same field of endeavor of creating and manipulating electronic documents. At the time the invention was made, it would have been obvious to one of ordinary skill in the art to differentiate between a simple element and a complex element, and to use such differentiation to determine how to render a word document. It is at least obvious, and possibly inherent, from the screenshot and the code example shown in Ayers that AbiWord determines whether the data is complex or simple.

The suggestion or motivation for combining the teachings of Ayers, that a word processor program could convert and store documents in XML, with the teaching of XML Schema, including that document types are distinguished as simple or complex, is expressly stated in XML Requirements, stating: "The following usage scenarios describe XML applications that should benefit from XML schemas. * * * 4. Traditional document authoring/editing governed by schema constraints." See, XML Requirements, page 3.

The combination of the teachings of Ayers to convert and store word processing documents in XML combined with the teachings of SML Schema that data may be one of two types, simple or complex, would be an invention whereby a document mapped to XML would also distinguish and map simple and complex elements, and that such could be stored in memory.)

writing the properties into at least one of a markup language element, an attribute, and a value; and
(See, Ayers, page 3, last two full paragraphs and citing code example from the screenshot and showing font and line properties and values. Note that Ayers teaches that the code is the markup language mapped from the properties of the document field displayed in the screenshot.)

storing the properties in the markup language document such that the fields of the word-processing document are substantially maintained when the markup language document is parsed by an application.

(See, Ayers, page 3, last two full paragraphs and citing code example from the screenshot. Note that Ayers teaches saving the document shown in the screenshot, which creates the saved code that is saved in memory.)

Regarding **dependent claim 12**, claim 12 incorporates substantially similar subject matter as claimed in claim 9, and is rejected along the same rationale.

Regarding **dependent claim 13**, claim 13 incorporates substantially similar subject matter as claimed in claim 10, and is rejected along the same rationale.

Regarding **dependent claim 14**, claim 14 incorporates substantially similar subject matter as claimed in claim 3, and is rejected along the same rationale.

Regarding **dependent claim 15**, claim 15 incorporates substantially similar subject matter as claimed in claim 4, and is rejected along the same rationale.

Regarding **dependent claim 16**, claim 16 incorporates substantially similar subject matter as claimed in claim 5, and is rejected along the same rationale.

Regarding **dependent claim 17**, claim 17 incorporates substantially similar subject matter as claimed in claim 6, and is rejected along the same rationale.

Regarding **dependent claim 18**, claim 18 incorporates substantially similar subject matter as claimed in claim 7, and is rejected along the same rationale.

Regarding **independent claim 19**, Ayers in view of XML Schema and further in view of XML Requirements teaches:

*A system for representing fields in a markup language document, comprising:
an application that is configured to determine properties relating to a field included in at least one section of an application document;
determine whether the field is one of a complex field and a simple field;
map the properties into at least one of a markup language element, an attribute, and a value; and
store the properties in the markup language document; and
a validation engine configured to validate the markup language document.*

(Claim 19 incorporates substantially similar subject matter as claimed in claim 11, and in further view of the following, is rejected along the same rationale.

Validation of an XML document is expressly taught in XML Schema, which states: "An instance document may be processed against a schema to verify whether the rules specified in the schema are honored in the instance. Typically, such processing actually does two things, (1) it checks for conformance to the rules, a process called schema validation . . ." See, XML Schema, downloaded page 59. See also, XML Schema, downloaded page 60, expressly teaching validation of simple and

complex element types. It would have been obvious to one of ordinary skill in the art at the time of the invention to validate an instance document in order to confirm that it follows the rules of the schema.)

Regarding **dependent claim 20**, claim 20 incorporates substantially similar subject matter as claimed in claim 9, and is rejected along the same rationale.

Regarding **dependent claim 21**, claim 21 incorporates substantially similar subject matter as claimed in claim 3, and is rejected along the same rationale.

Regarding **dependent claim 22**, claim 22 incorporates substantially similar subject matter as claimed in claim 4, and is rejected along the same rationale.

Regarding **dependent claim 23**, claim 23 incorporates substantially similar subject matter as claimed in claim 10, and is rejected along the same rationale.

25. It is noted that any citations to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art.

See, MPEP 2123.

Conclusion

26. The following prior art is made of record and not relied upon that is considered pertinent to applicants' disclosure:

Ozzie, et al. (U.S. Patent 6,941,510 B1), teaching storage of XML compliant documents.

Black, et al. (U.S Patent 6,763,500 B2), teaching validation and merging of electronic documents.

Friedman (U.S. Patent 6,675,353 B1), teaching generating XML documents without generating hierarchical tree structures.

Fittges, et al. (U.S. Patent 6,754,648 B1), teaching XML database.

Ray, Erik T., "Learning XML," O'Reilly & Associates, Inc., January 2001, cover, copyright, and Chapter 5, downloaded pages 1-25.

Glenn, Walter, "Word 2000 in a Nutshell," O'Reilly & Associates, Inc., August 2000, cover, copyright, and sections 16.4 and 16.3, downloaded pages 1-8.

Liberty, J. and Kraley, M., "XML Web Documents from Scratch," Que Corporation, March 10, 2000, cover, copyright, chapters 1-2, downloaded pages 1-16.

Mosely, L.E., and Boodey, D.M., "Mastering Microsoft Office 97, Professional Edition," 1996, cover, copyright, pages 87, 94-98, 103-105, 165-179, and 1114-1115.

Watchorn, H. and Daly, P., "Word and XML: Making the 'Twain Meet,'" XML Europe 2001, papers, May 2001, downloaded pages 1-11.

Novak, U., et al. "Experimental XSLT Processor for Objects," Proceedings of the JASTED Int'l Conf. on Applied Informatics, February 2002, pages 277-282.

XML Workshop Ltd., "Word to XML Converters," March 7, 2003, downloaded pages 1-2.

Wen, H., "AbiWord: Open Source's Answer to Microsoft Word," Linux Devcenter.com, March 14, 2002, downloaded pages 1-3.

Alschuler, L., "Getting the Tags In: Vendors Grapple with XML-Authoring, Editing and Cleanup," The Seybold Report on Internet Publishing, Vol. 5, No. 6, February 2001, reprint by Hypervision, pages 1-6.

Schmelzer, R., "ZapThink Briefing Note: HyperVision Automating Valid XML Document Creation within Microsoft Word," ZapThink, February 18, 2002, pages 1-6.

YAWC Pro, "Welcome to YAWC Pro," December 11, 2001, 1 page.

"YAWC Pro 1.0 Installation & User Guide," pages 1-11.

"Case Study: Converting Word into XML," YAWC Pro, 1 page.

"Case Study: Maintaining Websites with Microsoft Word," YAWC Pro, 1 page.

"Case Study: Publishing Content to the Web and Mobile Phones," YAWC Pro, 1 page.

"Case Study: Typsetting XML with QuarkXPress," YAWC Pro, 1 page.

Sklar, D., "The Annotated Rainbow DTD, Rainbow version 2.5," Electronic Book Technologies, Inc., February 8, 1995, pages 1-12.

Tetrasix, "Welcome to Tetrasix Web Site," re: MajiX, April 18, 2001, downloaded pages 1-3.

Dzuba, V., "Majix 1.0: A Word to XML Converter," October 6, 1998, pages 1-2.
Infinity-Loop, Web Site Home Page, re: infinity-loop, April 20, 2001, 1 page.

Sun Microsystems, "The OpenOffice.org Source Project," Sun Microsystems, Inc., 2000, downloaded pages 1-34.

HyperVision, Ltd., "WorX 2.1 Authoring Guide for XML 2001," September 2001, pages 1-29.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael K. Botts whose telephone number is 571-272-5533. The examiner can normally be reached on Monday Thru Friday 8:00-4:00 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather Herndon can be reached on 571-272-4136. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MKB

William L. Bashore
WILLIAM BASHORE
PRIMARY EXAMINER

12/22/2005